REMARKS

Claims 1-26 are pending.

In the following, the Examiner's comments, when included, are presented in bold, indented type, followed by the Applicants' remarks.

DETAILED ACTION

Claims 1-26 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

The claimed invention is directed to non-statutory subject matter because the claim does not require any physical transformation, and the invention as claimed does not produce a useful, and tangible result in view of MPEP 2106 (IV)(C)(2)((B))((2))(a) and (b).

The office action does not make a *prima facie* case of unpatentability under 35 USC 101. To make such a case, the office action is required to "identify and explain in the record the reasons why a claim is for an abstract idea with no practical application. . . ." MPEP 2106(IV)(D). Only then does the burden shift to the applicant to "amend the claim or make a showing of why the claim is eligible for patent protection." *Id.* The office action's conclusory statement that "the claim does not produce a useful, [sic] and tangible result" is nothing more than a statement of the MPEP's test for utility. Applicant respectfully requests that this rejection be withdrawn.

Claim Rejections - 35 USC § 112

Claims 1, 11, 18 and 26 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

As in claim 1, the clause *the buffer* (line 6) references to other items in the claim. It is unclear what item is being referenced.

As in claim 11, the clause *the buffer* (line 6) references to other items in the claim. It is unclear what item is being referenced.

As in claim 18, the clause *the buffer* (line 6) references to other items in the claim. It is unclear what item is being referenced.

As in claim 26, the clause *the buffer* (line 14) references to other items in the claim. It is unclear what item is being referenced.

Applicant has amended claims 1, 11, 18 and 26.

Claim Rejections - 35 USC § 102

Claims 1-4, 6-20 and 22-25 are rejected under 35 U.S.C. 102(b) as being anticipated by Ross [USP 5,666,525].

Regarding claims 1 and 18, Ross teaches a method for redistributing data in a relational data base management system, comprising:

allocating a buffer associated with a transmitting processing module **(Col. 5 Lines 1-5 and Col. 6 Lines 13-18, available memory as *a buffer* is allocated, the available memory is associated with a write operation as *transmitting processing module*), *the transmitting processing module having access to a program, the program capable of managing a redistribution of one or more rows associated with one or more database tables* (Col. 6 Lines 22-29, the write operation having access to the flush operation as a program, the flush operation is *capable of managing a redistribution of one or more rows associated with one or more database tables*);**

comparing the allocated buffer to a portion of the buffer to be occupied by the one or more rows **(Col. 5 Lines 18-21);**

if the allocated buffer is larger than the portion of the buffer to be occupied by the one or more rows **(As disclosed by Ross at Col. 5 Lines 13-21, "the number of buffers is determined by the total number and size of unique selected records. If the total size of the records to be read from a second input table during a join of two tables is three times the size of the available memory, the memory will be partitioned into three buffers". In different words, if the total size of the records is less than the size of the available memory, the memory will not be partitioned):**

> ***storing one or more rows of a database table in the allocated buffer*** **(Col. 6 Lines 13-15, because the total size of the records is less than the size of the available memory, the *one or more rows of a database table* is written to the available memory);**
>
> ***communicating a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer*** **(Col. 6 Lines 30-45, the records from the**

available memory is flushed to disk as *message comprising at least some of the one or more rows stored in the allocated buffer* for storing in an output file. The storing operation is considered as *one or more destination processing modules*);

otherwise:

executing a many-rows method to redistribute the one or more rows (As disclosed by Ross at Col. 5 Lines 13-21, the partitioning method is considered as *a many-rows method* to *redistribute the one or more rows* as disclosed from Col. 5 Line 60 to Col. 6 Line 45).

Regarding claims 1 and 18, Ross teaches a method for redistributing data in a relational data base management system, comprising:

allocating a buffer associated with a transmitting processing module (Col. 5 Lines 1-5 and Col. 6 Lines 13-18, available memory as *a buffer* is allocated, the available memory is associated with a write operation as *transmitting processing module*), *the transmitting processing module having access to a program, the program capable of managing a redistribution of one or more rows associated with one or more database tables* (Col. 6 Lines 22-29, the write operation having access to the flush operation as a program, the flush operation is *capable of managing a redistribution of one or more rows associated with one or more database tables*);

comparing the allocated buffer to a portion of the buffer to be occupied by the one or more rows (Col. 5 Lines 18-21);

if the allocated buffer is larger than the portion of the buffer to be occupied by the one or more rows :

storing one or more rows of a database table in the allocated buffer ;

communicating a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer (Because the available memory is smaller than the total size of record, the method go to otherwise condition);

otherwise:

executing a many-rows method to redistribute the one or more rows (As disclosed by Ross at Col. 5 Lines 13-21, the partitioning method is considered as *a many-rows method* to *redistribute the one or more rows* as disclosed from Col. 5 Line 60 to Col. 6 Line 45).

Applicant respectfully disagrees. The office action appears to equate Ross's available main memory (Ross, col. 5, lines 11-13) with the allocated buffer required by claims 1, 11, 18, and 26. However, Ross does not teach or suggest performing different actions depending on whether the allocated buffer is larger than the one or more rows to be redistributed, as required by claims 1, 11, 18, and 26. Ross describes flushing a buffer to a mass storage disk if available allocated memory for that buffer becomes full, Ross col. 6, lines 22-25, and flushing records from an input table to main memory, Ross col. 6, lines 34-38, but does not teach or suggest performing one process if the allocated buffer (i.e., the "available main memory" according to this interpretation of the office action) is larger than the one or more rows to be redistributed and otherwise performing another process, as required by claims 1, 11, 18, and 26.

If the office action is equating Ross's buffers with the allocated buffer required by claims 1, 11, 18, and 26, then Ross does not teach or suggest a many-rows method to redistribute the one or more rows as required by claims 1, 11, 18, and 26, which the office action equates with Ross's partitioning method for distributing records from tables into buffers, Ross, col. 5, line 60 – col. 6, line 45. If the office action is equating Ross's buffers with the recipient of rows distributed through the many-rows method, Ross's buffers cannot be both the allocated buffer required by claims 1, 11, 18, and 26 and the recipient of rows distributed from the allocated buffer through the many-rows method of redistributing the one or more rows, as required by claims 1, 11, 18, and 26.

If the office action is equating Ross's output tables with the recipients of rows distributed from the allocated buffer through the many-rows method of redistributing the one or more rows, as required by claims 1, 11, 18, and 26, then Ross is missing the requirement that the one or more rows be redistributed through a many-rows method if the allocated buffer is not larger than

the one or more rows, as required by amended claims 1, 11, 18, and 26. The office action equates the many-rows method with Ross's partitioning method. In the portion of Ross referred to in the office action as the partitioning method, data is distributed through Ross's buffers, which the office action (under this interpretation) equates with the allocated buffer required by the claims, to Ross's output tables, which the office action (under this interpretation) equates with the recipients of rows distributed from the allocated buffer through the many-rows method of redistributing the one or more rows. This is counter to what is required by claims 1, 11, 18 and 26, which require that, when the allocated buffer is not larger than the one or more rows, the one or more rows should be redistributed through a many-rows method, without first storing the one or more rows in the allocated buffer.

Further, Ross does not teach or suggest a many-rows method to redistribute the one or rows, as required by claims 1, 11, 18, and 26. Ross's partitioning method is not a many-rows redistribution method as suggested in the office action. An example of a many-rows method to distribute the one or more rows is described in Fig. 5 and accompanying text of the instant application. Ross does not teach or suggest such a method.

> **Regarding claims 10 and 25, Ross teaches a method and system for redistributing data in a relational data base management system, comprising:**
>
> ***invoking a program on one or more of a plurality of transmitting modules, the program capable of managing of one or more rows associated with one or more database tables*** **(Col. 4 Line 64-Col. 5 Line 1);**
>
> ***if the program was invoked on a single transmitting module; executing a few-rows redistribution method to redistribute the one or more rows*** **(As disclosed by Ross at Col. 5 Lines 13-21, "the number of buffers is determined by the total number and size of unique selected records. If the total size of the records to be read from a second input table during a join of two tables is three times the size of the available memory, the memory will be partitioned into three buffers". In different words, if the total size of the records is less than the size of the available memory, the memory will not be partitioned and be considered as *single transmitting module*. As disclosed at Col. 6 Lines**

13-15, because the total size of the records is less than the size of the available memory, the *one or more rows of a database table* is written to the available memory);

***otherwise; executing a many-rows redistribution method to redistribute the one or more rows* (As disclosed by Ross at Col. 5 Lines 13-21, the partitioning method is considered as *a many-rows method* to *redistribute the one or more rows* as disclosed from Col. 5 Line 60 to Col. 6 Line 6).**

Applicant respectfully disagrees. Ross does not teach or hint at executing a few-rows redistribution method to redistribute the one or more rows, as required by claims 10 and 25. An example of a few-rows distribution method is described in the specification in Fig. 6 and accompanying text. Ross does not teach or suggest such a method.

Further, Ross does not hint at executing a many-rows redistribution method to redistribute the one or more rows, as required by claims 10 and 25. Ross's partitioning method is not a many-rows redistribution method as suggested in the office action. An example of a many-rows method to distribute the one or more rows is described in Fig. 5 and accompanying text of the instant application. Ross does not teach or suggest such a method.

Regarding claims 2 and 12, Ross teaches all the claim subject matters as discussed above with respect to claims 1 and 11, Ross further discloses *the message comprises all of the one or more rows stored in the allocated buffer* (Col. 6 Lines 30-45).

Regarding claims 3, 13 and 19, Ross teaches all the claim subject matters as discussed above with respect to claims 1, 11 and 18, Ross further discloses *the transmitting processing module comprises one of a plurality of processing modules associated with a relational database system* (Col. 6 Lines 13-18).

Regarding claims 4, 14 and 20, Ross teaches all the claim subject matters as discussed above with respect to claims 1, 11 and 18, Ross further discloses *the message is communicated to each of a plurality of destination processing elements* (Col. 6 Lines 30-45).

Regarding claims 6, 16 and 22, Ross teaches all the claim subject matters as discussed above with respect to claims 1, 10 and 18, Ross further discloses *the many-rows method comprises*:

communicating from one or more transmitting processing modules a first signal to a plurality of processing modules within a relational database system, the first signal operable to initiate a row receiver task on each of the processing modules; from one or more of the processing modules a ready-to-receive signal to the one or more transmitting processing modules; communicating from the one or more transmitting processing modules a second signal comprising the one or more rows associated with the database table; after communication of the last row associated with the database table, communicating from the one or more transmitting processing modules an end-of-data signal to each of the plurality of processing modules (Col. 5 Line 60 to Col. 6 Line 45).

Regarding claims 7 and 23, Ross teaches all the claim subject matters as discussed above with respect to claims 1 and 18, Ross further discloses *invoking the program on a single transmitting processing module* (As disclosed by Ross at Col. 5 Lines 13-21, if the total size of the records is less than the size of the available memory, the memory will not be partitioned and be considered as *single transmitting module*).

Regarding claim 8, Ross teaches all the claim subject matters as discussed above with respect to claim 1, Ross further discloses the step of *receiving at each of a plurality of destination processing elements a substantially similar set of the one or more rows stored in the allocated buffer* (Col. 6 Lines 30-45).

Regarding claims 9, 15 and 24, Ross teaches all the claim subject matters as discussed above with respect to claims 1, 11 and 18, Ross further discloses the step of *determining a number of rows to store in the allocated buffer* (Col. 5 Lines 1-21).

Regarding claim 11, Ross teaches all the claim subject matters as discussed above with respect to claim 10, Ross further discloses the steps of

allocating a buffer associated with a transmitting processing module (Col. 5 Lines 1-5 and Col. 6 Lines 13-18, available memory as *a buffer* is allocated, the available memory is associated with a write operation as *transmitting processing module*), *the transmitting processing module having access to a program, the program capable of managing a redistribution of one or more rows associated with one or more database tables* (Col. 6 Lines 22-29, the write operation having access to the flush operation as a program, the flush operation is *capable of managing a redistribution of one or more rows associated with one or more database tables*);

comparing the allocated buffer to a portion of the buffer to be occupied by the one or more rows (Col. 5 Lines 18-21);

if the allocated buffer is larger than the portion of the buffer to be occupied by the one or more rows (As disclosed by Ross at Col. 5 Lines 13-21,

"the number of buffers is determined by the total number and size of unique selected records. If the total size of the records to be read from a second input table during a join of two tables is three times the size of the available memory, the memory will be partitioned into three buffers". In different words, if the total size of the records is less than the size of the available memory, the memory will not be partitioned):

> ***storing one or more rows of a database table in the allocated buffer*** **(Col. 6 Lines 13-15, because the total size of the records is less than the size of the available memory, the *one or more rows of a database table* is written to the available memory);**
>
> ***communicating a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer*** **(Col. 6 Lines 30-45, the records from the available memory is flushed to disk as *message comprising at least some of the one or more rows stored in the allocated buffer* for storing in an output file. The storing operation is considered as *one or more destination processing modules*);**
>
> ***otherwise:***

executing a many-rows method to redistribute the one or more rows **(As disclosed by Ross at Col. 5 Lines 13-21, the partitioning method is considered as a many-rows method to *redistribute the one or more rows* as disclosed from Col. 5 Line 60 to Col. 6 Line 45).**

Regarding claim 11, Ross teaches all the claim subject matters as discussed above with respect to claim 10, Ross further discloses the steps of:

allocating a buffer associated with a transmitting processing module **(Col. 5 Lines 1-5 and Col. 6 Lines 13-18, available memory as a buffer is allocated, the available memory is associated with a write operation as *transmitting processing module*), *the transmitting processing module having access to a program, the program capable of managing a redistribution of one or more rows associated with one or more database tables*** **(Col. 6 Lines 22-29, the write operation having access to the flush operation as a program, the flush operation is *capable of managing a redistribution of one or more rows associated with one or more database tables*);**

comparing the allocated buffer to a portion of the buffer to be occupied by the one or more rows **(Col. 5 Lines 18-21);**

if the allocated buffer is larger than the portion of the buffer to be occupied by the one or more rows :

> ***storing one or more rows of a database table in the allocated buffer ;***

communicating a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer **(Because the available memory is smaller than the total size of record, the method go to otherwise condition);**

otherwise:

executing a many-rows method to redistribute the one or more rows **(As disclosed by Ross at Col. 5 Lines 13-21, the partitioning method is considered as *a many-rows method* to *redistribute the one or more rows* as disclosed from Col. 5 Line 60 to Col. 6 Line 45).**

Regarding claim 17, Ross teaches all the claim subject matters as discussed above with respect to claim 10, Ross further discloses the step of *determining the number of transmitting modules on which the program was invoked* (Col. 5 Lines 13-34).

The claims discussed above depend from one of claims 1, 10, 11, 18, 25 or 26 and are patentable for at least the reasons described above for those claims.

Claim Rejections – 35 USC § 103

Claims 5 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ross [USP 5,666,525].

Regarding claims 5 and 21, Ross-teaches all the claim subject matters as discussed above with respect to claims 1 and 18, Ross does not explicitly teach claimed limitation *the allocated buffer is capable of storing no more than ten (10) rows*. However, as disclosed at Col. 5 Lines 10-12, available memory is that portion which is not allocated to another function or to another user in a multi-user computer system. Thus, during a particular processing time, the available memory is reduced and *capable of storing no more than ten (10) rows*. It would have been obvious for one of ordinary skill in the art at the time the invention was made to include this feature into the available memory in order to partition the available memory.

Claims 5 and 21 depend from claims 1 and 18 and are patentable for at least the reasons described above for those claims.

Claim 26 is rejected under 35 U.S.C. 103(a) as being unpatentable over Leverenz [Oracle8i Concept] in view of Ross [USP 5,666,525].

Regarding claim 26, Leverenz teaches a database management system (Page 17, Oracle Server), comprising:

a massively parallel processing system comprising (Page 53, Overview of Parallel Execution):

one or more nodes (Page 53, Overview of Parallel Execution, Oracle System);

a plurality of processors, each of the one or more nodes providing access to one or more processors (Page 53, Overview of Parallel Execution, Oracle System provides access to a plurality of CPUs); and

a plurality of virtual processes, each of the one or more processors providing access to one or more virtual processes (Page 54, Operation That Can Be Parallelized);

a set of one or more database tables residing on the one or more nodes (Page 61, FIG. 26-5, EMP Table); and

one or more of the plurality of virtual processes that allocate a buffer associated with a transmitting processing module (As shown and disclosed at Page 61 FIG. 26-5 and Page 58, How Parallel Execution Servers Communicate, scanning operations as *one or more of the plurality of virtual processes that allocate a buffer associated with* ORDER BY operations as *transmitting processing module*), *the transmitting processing module having access to a program, the program capable of managing a redistribution of one or more rows associated with one or more database tables* (As shown and disclosed at Page 61 FIG. 26-5, ORDER BY operations having access to Parallel Execution Coordinator as *a program.* The Parallel Execution Coordinator redistribute the query results include one or more rows to the user). The missing of Leverenz is the processes of *compare the allocated buffer to a portion of the buffer to be occupied by the one or more rows; if the allocated buffer is larger than the portion of the buffer to be occupied by the one or more rows: store one or more rows associated with a database table in the allocated buffer; communicate a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer; otherwise: execute a many-rows method to redistribute the one or more rows.*

Ross teaches a method for redistributing data in a relational data base management system, comprising:

comparing the allocated buffer to a portion of the buffer to be occupied by the one or more rows (Ross, Col. 5 Lines 18-21);

if the allocated buffer is larger than the portion of the buffer to be occupied by the one or more rows (As disclosed by Ross at Col. 5 Lines 13-21, "the number of buffers is determined by the total number and size of unique selected records. If the total size of the records to be read from a second input table during a join of two tables is three times the size of the available

memory, the memory will be partitioned into three buffers". In different words, if the total size of the records is less than the size of the available memory, the memory will not be partitioned):

> ***storing one or more rows of a database table in the allocated buffer*** **(Ross, Col. 6 Lines 13-15, because the total size of the records is less than the size of the available memory, the one or more rows of a database table is written to the available memory);**
>
> ***communicating a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer*** **(Ross, Col. 6 Lines 30-45, the records from the available memory is flushed to disk as *message comprising at least some of the one or more rows stored in the allocated buffer* for storing in an output file. The storing operation is considered as *one or more destination processing modules*);**
>
> ***otherwise:***

executing a many-rows method to redistribute the one or more rows **(As disclosed by Ross at Col. 5 Lines 13-21, the partitioning method is considered as *a many-rows method* to *redistribute the one or more rows* as disclosed from Col. 5 Line 60 to Col. 6 Line 45).**

The technique of partitioning the available is a must for Leverenz system in order to manage the available memory.

It would have been obvious for one of ordinary skill in the art at the time the invention was made to combine the teaching of Ross into the Leverenz in order to manage buffers during parallel processing.

Regarding claim 26, Leverenz teaches a database management system (Page 17, Oracle Server), comprising:

a massively parallel processing system comprising **(Page 53, Overview of Parallel Execution):**

one or more nodes **(Page 53, Overview of Parallel Execution, Oracle System);**

a plurality of processors, each of the one or more nodes providing access to one or more processors **(Page 53, Overview of Parallel Execution, Oracle System provides access to a plurality of CPUs); and**

a plurality of virtual processes, each of the one or more processors providing access to one or more virtual processes **(Page 54, Operation That Can Be Parallelized);**

a set of one or more database tables residing on the one or more nodes (Page 61, FIG. 26-5, EMP Table); and

one or more of the plurality of virtual processes that allocate a buffer associated with a transmitting processing module (As shown and disclosed at Page 61 FIG. 26-5 and Page 58, How Parallel Execution Servers Communicate, scanning operations as *one or more of the plurality of virtual processes that allocate a buffer associated with* ORDER BY operations as *transmitting processing module*), *the transmitting processing module having access to a program, the program capable of managing a redistribution of one or more rows associated with one or more database tables* (As shown and disclosed at Page 61 FIG. 26-5, ORDER BY operations having access to Parallel Execution Coordinator as a program. The Parallel Execution Coordinator redistribute the query results include one or more rows to the user). The missing of Leverenz is the processes of *compare the allocated buffer to a portion of the buffer to be occupied by the one or more rows; if the allocated buffer is larger than the portion of the buffer to be occupied by the one or more rows: store one or more rows associated with a database table in the allocated buffer; communicate a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer; otherwise: execute a many-rows method to redistribute the one or more rows.*

Ross teaches a method for redistributing data in a relational data base management system, comprising:

comparing the allocated buffer to a portion of the buffer to be occupied by the one or more rows (Col. 5 Lines 18-21);

if the allocated buffer is larger than the portion of the buffer to be occupied by the one or more rows :

storing one or more rows of a database table in the allocated buffer ;

communicating a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer (Because the available memory is smaller than the total size of record, the method go to otherwise condition);

otherwise:

executing a many-rows method to redistribute the one or more rows (As disclosed by Ross at Col. 5 Lines 13-21, the partitioning method is considered as *a many-rows method* to *redistribute the one or more rows* as disclosed from Col. 5 Line 60 to Col. 6 Line 45).

The technique of partioning the available is a must for Leverenz system in order to manage the available memory.

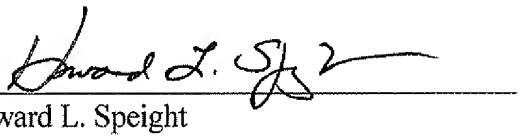
It would have been obvious for one of ordinary skill in the art at the time the invention was made to combine the teaching of Ross into the Leverenz in order to manage buffers during parallel processing.

Claim 26 is patentable for at least the reasons described above for claims 1, 11, 18 and 26.

SUMMARY

Applicant contends that the claims are in condition for allowance, which action is requested. Applicant does not believe any fees are necessary with the submitting of this response. Should any fees be required, Applicant requests that the fees be debited from deposit account number 14-0225, Order Number 11235.

Respectfully submitted,



Howard L. Speight
Reg. No. 37,733
9601 Katy Freeway
Suite 280
Houston, Texas 77024
(713) 881-9600 (phone)
(713) 715-7384 (facsimile)
howard@hspeight.com
ATTORNEY FOR APPLICANTS

Date: February 14, 2007